# Newsboat HPR Download Script — Conversation Export

Exported from Claude conversation — May 2026

### USER

```
wget `curl https://tuxjam.otherside.network/feed/podcast/ | grep -o 'https*:/
```

But instead of only getting one episode, download the latest 5

### ASSISTANT

```
curl https://tuxjam.otherside.network/feed/podcast/ | grep -o 'https*://[^"]*
```

`wget -i -` reads URLs from stdin, so piping the 5 URLs directly into it grabs all of them in one shot.

### USER

Rewrite this instead

```
wget -nc --content-disposition -P ~/podcasts/hub.hackerpublicradio.org/HPR-ne
```

which does not act like expected, but downloads all shows in the feed. I would like to get n shows, in this case 2 (signified by head -2)

### ASSISTANT

The issue is that `xargs` joins all URLs into one line before `head -2` can limit them — so all URLs get passed to `wget` at once.

```
curl http://hackerpublicradio.org/hpr_ogg_rss.php | grep -Eo 'https?://[^"]*\
```

Key fixes:

- Removed the premature `xargs` before `head -2`, so `head` actually limits the list to 2 URLs
- Moved `xargs wget ...` to the end, after the list is already trimmed
- Cleaned up `https*` → `https?` (the original would also match `httpssss`, etc.)

**USER**

Get input from `http://hackerpublicradio.org/hpr_ogg_rss.php` — "~HPR-two-week-feed" (HPR) (Podcast) when in Newsboat, and use it for a macro to download only a single ogg file from a feed entry

**ASSISTANT**

In your Newsboat `config` file:

```
macro d set browser "curl %u | grep -Eo 'https?://[^\"]*\\.ogg' | head -1 | x
```

Then in Newsboat, navigate to the HPR feed entry and press `,d` (comma = macro prefix, d = your macro key).

- The macro temporarily swaps `browser` to your download command, calls `open-in-browser` (which passes the entry URL as `%u`), then restores your normal browser
- `head -1` grabs only the single `.ogg` from that episode's page
- Adjust `-P ~/podcasts/hpr/` to wherever you want the files saved
- Replace `"your-normal-browser"` with whatever you normally have set, e.g. `"firefox"` or `"w3m %u"`

**USER**

```
curl: (3) URL rejected: Bad hostname
wget: missing URL
Usage: wget [OPTION]... [URL]...
Try `wget --help' for more options.
Cleaning up cache...done.
```

**ASSISTANT**

The issue is that `%u` in Newsboat macros gives you the entry's *webpage* URL, not the feed's enclosure/audio URL. Try fetching the individual episode page and scraping the ogg link from it:

```
macro d set browser "wget -nc --content-disposition -P ~/podcasts/hpr/ $(curl
```

If that still fails, verify what URL Newsboat is actually passing:

```
macro t set browser "echo %u > /tmp/newsboat-url.txt"; open-in-browser ; set
```

Then check `cat /tmp/newsboat-url.txt`.

**USER**

For reference, the test download path is `~/podcasts/hub.hackerpublicradio.org/HPR-newsboat-test/`

**ASSISTANT**

Updated with your test path. What does `cat /tmp/newsboat-url.txt` show after running the test macro on an HPR entry?

**USER**

```
mark@mark-aspirea51543:~ $ cat /tmp/newsboat-url.txt
https://hackerpublicradio.org/eps/hpr4627/index.html
```

```
mark@mark-aspirea51543:~ $
```

but still fails to grab URL

**ASSISTANT**

Good — so `%u` gives a valid episode page URL. Let's test the curl scrape directly:

```
curl -s https://hackerpublicradio.org/eps/hpr4627/index.html | grep -Eo 'http
```

Or dump raw HTML to inspect:

```
curl -s https://hackerpublicradio.org/eps/hpr4627/index.html | grep -i ogg
```

**USER**

```
mark@mark-aspirea51543:~ $ curl -s https://hackerpublicradio.org/eps/hpr4627/
https://hub.hackerpublicradio.org/ccdn.php?filename=/eps/hpr4627/hpr4627.ogg
https://hub.hackerpublicradio.org/ccdn.php?filename=/eps/hpr4627/hpr4627.ogg
mark@mark-aspirea51543:~ $
```

**ASSISTANT**

The URL contains a query string. Test wget with the URL quoted to prevent shell interpretation of `?filename=...`:

```
wget -nc --content-disposition -P ~/podcasts/hub.hackerpublicradio.org/HPR-ne
```

Then update your macro with the URL properly quoted:

```
macro d set browser "wget -nc --content-disposition -P ~/podcasts/hub.hackerp
```

**USER**

That worked on the first couple of entries. Scripts used:

```
#!/bin/bash
# bin/download-and-rename-hpr-0.2.sh
wget -nc --content-disposition -P ~/podcasts/hub.hackerpublicradio.org/HPR-ne
cd ~/podcasts/hub.hackerpublicradio.org/HPR-newsboat-test/
exif-rename-hpr.sh
```

```
#!/bin/bash
# exif-rename-hpr-v0.3.sh
shopt -s nullglob
mp3s=(*.mp3)
oggs=(*.ogg)
if [[ ${#mp3s[@]} -gt 0 ]]; then
    podcast="mp3"
elif [[ ${#oggs[@]} -gt 0 ]]; then
    podcast="ogg"
else
    echo "No .mp3 or .ogg files found in $(pwd)"
    exit 1
fi
case $podcast in
    mp3)
        for i in *.mp3; do
          [[ "$i" =~ ^hpr[0-9]{4}\.mp3$ ]] || { echo "Skipping unexpected fil
            file="$(exiftool -p '$FileName ; ${Title}.$FileTypeExtension' "$i
            [[ -z "$file" ]] && { echo "exiftool failed on $i, skipping"; cor
            mv "$i" "$file"
            echo "$file"
        done
        ;;
    ogg)
        for i in *.ogg; do
          [[ "$i" =~ ^hpr[0-9]{4}\.ogg$ ]] || { echo "Skipping unexpected fil
            file="$(exiftool -p '$FileName ; ${Title}.$FileTypeExtension' "$i
            [[ -z "$file" ]] && { echo "exiftool failed on $i, skipping"; cor
            mv "$i" "$file"
            echo "$file"
        done
        ;;
    *)
        echo "Unknown file type."
```

```
        ;;
esac
```

**ASSISTANT**

Good progress. Update `download-and-rename-hpr.sh` to accept the URL as an argument:

```
#!/bin/bash
# ~/bin/download-and-rename-hpr.sh

URL="$1"

wget -nc --content-disposition -P ~/podcasts/hub.hackerpublicradio.org/HPR-ne
    "$(curl -s "$URL" | grep -Eo 'https?://[^"]*\.ogg' | head -1)"

cd ~/podcasts/hub.hackerpublicradio.org/HPR-newsboat-test/
~/bin/exif-rename-hpr.sh
```

Newsboat config:

```
macro d set browser "~/bin/download-and-rename-hpr.sh %u"; open-in-browser ;
```

Make sure the script is executable: `chmod +x ~/bin/download-and-rename-hpr.sh`

**USER**

```
Resolving "" ("")... failed: Name or service not known.
wget: unable to resolve host address '""'
Skipping unexpected file: hpr4618 ; Simple Podcasting - Episode 2 - Basic Fil
...
Prepended http:// to '""'
--2026-04-28 10:08:12--  http://%22%22/
Resolving "" ("")... failed: Name or service not known.
```

**ASSISTANT**

`$URL` is empty — `%u` isn't being expanded when passed as a script argument. Add debug logging:

```
#!/bin/bash
# ~/bin/download-and-rename-hpr.sh

URL="$1"
echo "DEBUG URL: $URL" >> /tmp/hpr-debug.log

OGG_URL="$(curl -s "$URL" | grep -Eo 'https?://[^"]*\.ogg' | head -1)"
echo "DEBUG OGG: $OGG_URL" >> /tmp/hpr-debug.log

if [[ -z "$OGG_URL" ]]; then
    echo "ERROR: Could not find ogg URL from: $URL" >> /tmp/hpr-debug.log
    exit 1
fi

wget -nc --content-disposition -P ~/podcasts/hub.hackerpublicradio.org/HPR-ne

cd ~/podcasts/hub.hackerpublicradio.org/HPR-newsboat-test/
~/bin/exif-rename-hpr.sh
```

Then: `cat /tmp/hpr-debug.log`

**USER**

```
mark@mark-aspirea51543:~ $ cat /tmp/hpr-debug.log
DEBUG URL:
DEBUG OGG:
ERROR: Could not find ogg URL from:
DEBUG URL:
...
```

**ASSISTANT**

`$URL` is empty — Newsboat only expands `%u` directly in the `browser` string, not in script arguments. Fix: write the URL to a temp file from the macro, then read it in the script.

Newsboat config:

```
macro d set browser "echo %u > /tmp/hpr-url.txt && ~/bin/download-and-rename-
```

Updated script:

```
#!/bin/bash
# ~/bin/download-and-rename-hpr.sh

URL="$(cat /tmp/hpr-url.txt)"
echo "DEBUG URL: $URL" >> /tmp/hpr-debug.log

OGG_URL="$(curl -s "$URL" | grep -Eo 'https?://[^"]*\.ogg' | head -1)"
echo "DEBUG OGG: $OGG_URL" >> /tmp/hpr-debug.log

if [[ -z "$OGG_URL" ]]; then
    echo "ERROR: Could not find ogg URL from: $URL" >> /tmp/hpr-debug.log
    exit 1
fi

wget -nc --content-disposition -P ~/podcasts/hub.hackerpublicradio.org/HPR-ne

cd ~/podcasts/hub.hackerpublicradio.org/HPR-newsboat-test/
~/bin/exif-rename-hpr.sh
```

**USER**

1. Include mp3 files to the last script
2. Add downloaded file to `.local/share/newsboat/queue`

**ASSISTANT**

```
#!/bin/bash
# ~/bin/download-and-rename-hpr.sh

URL="$(cat /tmp/hpr-url.txt)"
echo "DEBUG URL: $URL" >> /tmp/hpr-debug.log

OGG_URL="$(curl -s "$URL" | grep -Eo 'https?://[^"]*\.(ogg|mp3)' | head -1)"
echo "DEBUG OGG: $OGG_URL" >> /tmp/hpr-debug.log

if [[ -z "$OGG_URL" ]]; then
```

```
    echo "ERROR: Could not find audio URL from: $URL" >> /tmp/hpr-debug.log
    exit 1
fi

DEST=~/podcasts/hub.hackerpublicradio.org/HPR-newsboat-test/

wget -nc --content-disposition -P "$DEST" "$OGG_URL"

# Add to newsboat queue
echo "$OGG_URL $DEST" >> ~/.local/share/newsboat/queue

cd "$DEST"
~/bin/exif-rename-hpr.sh
```

**USER**

Yes, set the queue entry to reflect the post-rename filename instead.

**ASSISTANT**

```
#!/bin/bash
# ~/bin/download-and-rename-hpr.sh

URL="$(cat /tmp/hpr-url.txt)"
echo "DEBUG URL: $URL" >> /tmp/hpr-debug.log

AUDIO_URL="$(curl -s "$URL" | grep -Eo 'https?://[^"]*\.(ogg|mp3)' | head -1)"
echo "DEBUG AUDIO: $AUDIO_URL" >> /tmp/hpr-debug.log

if [[ -z "$AUDIO_URL" ]]; then
    echo "ERROR: Could not find audio URL from: $URL" >> /tmp/hpr-debug.log
    exit 1
fi

DEST=~/podcasts/hub.hackerpublicradio.org/HPR-newsboat-test/

DOWNLOADED="$(wget -nc --content-disposition -P "$DEST" "$AUDIO_URL" 2>&1 \
    | grep -Eo "Saving to: '.*'" | sed "s/Saving to: '//;s/'$//")"

echo "DEBUG DOWNLOADED: $DOWNLOADED" >> /tmp/hpr-debug.log

cd "$DEST"
```

```
~/bin/exif-rename-hpr.sh

RENAMED="$(ls -t "$DEST"*.{ogg,mp3} 2>/dev/null | head -1)"
echo "DEBUG RENAMED: $RENAMED" >> /tmp/hpr-debug.log

if [[ -n "$RENAMED" ]]; then
    echo "$AUDIO_URL $RENAMED" >> ~/.local/share/newsboat/queue
else
    echo "WARN: Could not determine renamed file" >> /tmp/hpr-debug.log
    echo "$AUDIO_URL" >> ~/.local/share/newsboat/queue
fi
```

**USER**

1. The renamed file is not in the queue, only `hpr4542.ogg`
2. Queue tail shows existing entries use format: `"url" "path" downloaded`

```
https://hub.hackerpublicradio.org/ccdn.php?filename=/eps/hpr4625/hpr4625.ogg
https://hub.hackerpublicradio.org/ccdn.php?filename=/eps/hpr4637/hpr4637.mp3
https://hub.hackerpublicradio.org/ccdn.php?filename=/eps/hpr4542/hpr4542.ogg
```

**ASSISTANT**

Fix: use `comm -13` on sorted before/after file lists to reliably identify the new renamed file, and match the queue format `"url" "path" downloaded`:

```
#!/bin/bash
# ~/bin/download-and-rename-hpr.sh

URL="$(cat /tmp/hpr-url.txt)"
echo "DEBUG URL: $URL" >> /tmp/hpr-debug.log

AUDIO_URL="$(curl -s "$URL" | grep -Eo 'https?://[^"]*\.(ogg|mp3)' | head -1)
echo "DEBUG AUDIO: $AUDIO_URL" >> /tmp/hpr-debug.log

if [[ -z "$AUDIO_URL" ]]; then
    echo "ERROR: Could not find audio URL from: $URL" >> /tmp/hpr-debug.log
    exit 1
fi

DEST=~/podcasts/hub.hackerpublicradio.org/HPR-newsboat-test/
```

```
BEFORE="$(ls "$DEST"*.{ogg,mp3} 2>/dev/null | sort)"
wget -nc --content-disposition -P "$DEST" "$AUDIO_URL"
cd "$DEST"

AFTER="$(ls "$DEST"*.{ogg,mp3} 2>/dev/null | sort)"
DOWNLOADED="$(comm -13 <(echo "$BEFORE") <(echo "$AFTER"))"
echo "DEBUG DOWNLOADED: $DOWNLOADED" >> /tmp/hpr-debug.log

~/bin/exif-rename-hpr.sh

AFTER_RENAME="$(ls "$DEST"*.{ogg,mp3} 2>/dev/null | sort)"
RENAMED="$(comm -13 <(echo "$BEFORE") <(echo "$AFTER_RENAME"))"
echo "DEBUG RENAMED: $RENAMED" >> /tmp/hpr-debug.log

if [[ -n "$RENAMED" ]]; then
    echo "\"$AUDIO_URL\" \"$RENAMED\" downloaded" >> ~/.local/share/newsboat/
else
    echo "WARN: Could not determine renamed file" >> /tmp/hpr-debug.log
    echo "\"$AUDIO_URL\" \"$DOWNLOADED\" downloaded" >> ~/.local/share/newsbo
fi
```

#### USER

This works reliably. Sort the downloads into these locations for Hacker Public Radio:

- `http://hackerpublicradio.org/hpr_ogg_rss.php` `"~HPR-two-week-feed"` (HPR) (Podcast)
- `http://hackerpublicradio.org/rss-future.php` `"~HPR-future-feed"` (HPR) (Podcast)
- `https://hackerpublicradio.org/hpr_total_ogg_rss.php` `"~HPR-entire-feed"` (HPR) (Podcast)

#### ASSISTANT

Three macros, one per feed, writing a feed identifier to `/tmp/hpr-feed.txt`:

```
macro 1 set browser "echo %u > /tmp/hpr-url.txt && echo two-week > /tmp/hpr-f
macro 2 set browser "echo %u > /tmp/hpr-url.txt && echo future > /tmp/hpr-fee
```

```
macro 3 set browser "echo %u > /tmp/hpr-url.txt && echo entire > /tmp/hpr-fee
```

Script with case routing:

```
#!/bin/bash
URL="$(cat /tmp/hpr-url.txt)"
FEED="$(cat /tmp/hpr-feed.txt)"
AUDIO_URL="$(curl -s "$URL" | grep -Eo 'https?://[^"]*\.(ogg|mp3)' | head -1)

if [[ -z "$AUDIO_URL" ]]; then exit 1; fi

case "$FEED" in
    two-week) DEST=~/podcasts/hub.hackerpublicradio.org/HPR-two-week-feed/ ;;
    future)   DEST=~/podcasts/hub.hackerpublicradio.org/HPR-future-feed/ ;;
    entire)   DEST=~/podcasts/hub.hackerpublicradio.org/HPR-entire-feed/ ;;
    *)        DEST=~/podcasts/hub.hackerpublicradio.org/HPR-newsboat-test/ ;;
esac

mkdir -p "$DEST"
BEFORE="$(ls "$DEST"*.{ogg,mp3} 2>/dev/null | sort)"
wget -nc --content-disposition -P "$DEST" "$AUDIO_URL"
cd "$DEST"
AFTER="$(ls "$DEST"*.{ogg,mp3} 2>/dev/null | sort)"
DOWNLOADED="$(comm -13 <(echo "$BEFORE") <(echo "$AFTER"))"
~/bin/exif-rename-hpr.sh
AFTER_RENAME="$(ls "$DEST"*.{ogg,mp3} 2>/dev/null | sort)"
RENAMED="$(comm -13 <(echo "$BEFORE") <(echo "$AFTER_RENAME"))"

if [[ -n "$RENAMED" ]]; then
    echo "\"$AUDIO_URL\" \"$RENAMED\" downloaded" >> ~/.local/share/newsboat/
else
    echo "\"$AUDIO_URL\" \"$DOWNLOADED\" downloaded" >> ~/.local/share/newsbo
fi
```

**USER**

Instead, grep for line item in `~/.config/newsboat/urls` such as `http://hackerpublicradio.org/hpr_ogg_rss.php` and download it to `~/podcasts/hub.hackerpublicradio.org/HPR-two-week-feed/`

**ASSISTANT**

Use a single macro capturing both `%u` and `%F` (feed URL), then look up the `~tag` from the urls file:

Newsboat config:

```
macro d set browser "echo %u > /tmp/hpr-url.txt && echo %F > /tmp/hpr-feed-ur
```

Script:

```
#!/bin/bash
# ~/bin/download-and-rename-hpr.sh

URL="$(cat /tmp/hpr-url.txt)"
FEED_URL="$(cat /tmp/hpr-feed-url.txt)"
echo "DEBUG URL: $URL" >> /tmp/hpr-debug.log
echo "DEBUG FEED_URL: $FEED_URL" >> /tmp/hpr-debug.log

AUDIO_URL="$(curl -s "$URL" | grep -Eo 'https?://[^"]*\.(ogg|mp3)' | head -1)

if [[ -z "$AUDIO_URL" ]]; then
    echo "ERROR: Could not find audio URL from: $URL" >> /tmp/hpr-debug.log
    exit 1
fi

BASE=~/podcasts/hub.hackerpublicradio.org
DEST="$BASE/HPR-newsboat-test/"  # fallback

while IFS= read -r line; do
    [[ "$line" =~ ^# ]] && continue
    [[ -z "$line" ]] && continue
    feed="$(echo "$line" | awk '{print $1}')"
    if [[ "$feed" == "$FEED_URL" ]]; then
        tag="$(echo "$line" | grep -Eo '~[A-Za-z0-9_-]+' | head -1)"
        if [[ -n "$tag" ]]; then
            DEST="$BASE/${tag#\~}/"
        fi
        break
    fi
done < ~/.config/newsboat/urls

echo "DEBUG DEST: $DEST" >> /tmp/hpr-debug.log
mkdir -p "$DEST"

BEFORE="$(ls "$DEST"*.{ogg,mp3} 2>/dev/null | sort)"
wget -nc --content-disposition -P "$DEST" "$AUDIO_URL"
```

```
cd "$DEST"
AFTER="$(ls "$DEST"*.{ogg,mp3} 2>/dev/null | sort)"
DOWNLOADED="$(comm -13 <(echo "$BEFORE") <(echo "$AFTER"))"
~/bin/exif-rename-hpr.sh
AFTER_RENAME="$(ls "$DEST"*.{ogg,mp3} 2>/dev/null | sort)"
RENAMED="$(comm -13 <(echo "$BEFORE") <(echo "$AFTER_RENAME"))"

if [[ -n "$RENAMED" ]]; then
    echo "\"$AUDIO_URL\" \"$RENAMED\" downloaded" >> ~/.local/share/newsboat/
else
    echo "\"$AUDIO_URL\" \"$DOWNLOADED\" downloaded" >> ~/.local/share/newsbo
fi
```

First confirm `%F` works:

```
macro t set browser "echo %F > /tmp/hpr-feed-url.txt"; open-in-browser ; set
```

Then `cat /tmp/hpr-feed-url.txt` to verify.